**SET B- PART B**

**1. Elaborate on the instruction sequencing steps required for executing the following statements in the CPU with the help of the register set.**

**MOV AL, A**

**MOV BL, B**

**ADD AL, BL**

* Instruction execution needs the following steps, which are
* PC (program counter) register of the processor gives the address of the instruction which needs to be fetched from the memory.
* If the instruction is fetched then, the instruction opcode is decoded. On decoding, the processor identifies the number of operands. If there is any operand to be fetched from the memory, then that operand address is calculated.
* Operands are fetched from the memory. If there is more than one operand, then the operand fetching process may be repeated (i.e. address calculation and fetching operands).
* After this, the data operation is performed on the operands, and a result is generated.
* If the result has to be stored in a register, the instructions end here.

**The register movement needs to be shown for each instruction**

**2. Calculate the effective address for the following register:  
SS: 3860H, SP: 1735H, BP: 4826H**

(SS X 10H) + SP = 3640H X 10H + 1735H

= 36400H + 1735H

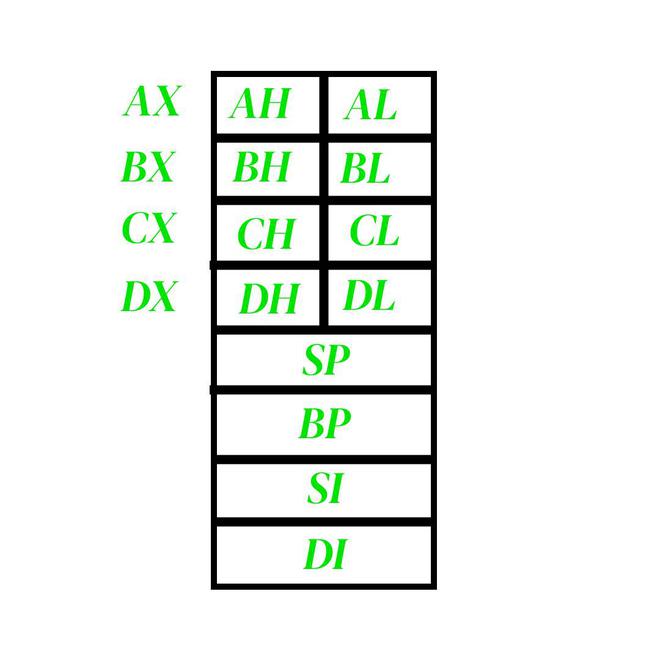
= 38135H

(SS X 10H) + BP = 3640H X 10H + 4826H

= 36400H + 4826H

= 41226H

**3. Categorize the registers of the 8086 microprocessor and discuss them in detail with a neat sketch.**



**1. AX:** This is the accumulator. It is of 16 bits and is divided into two 8-bit registers AH and AL to also perform 8-bit instructions. It is generally used for arithmetical and logical instructions but in 8086 microprocessor it is not mandatory to have an accumulator as the destination operand. Example:

ADD AX, AX (AX = AX + AX)

**2. BX:**This is the base register. It is of 16 bits and is divided into two 8-bit registers BH and BL to also perform 8-bit instructions. It is used to store the value of the offset. Example:

MOV BL, [500] (BL = 500H)

**3. CX:** This is the counter register. It is of 16 bits and is divided into two 8-bit registers CH and CL to also perform 8-bit instructions. It is used in looping and rotation. Example:

MOV CX, 0005

LOOP

**4. DX:** This is the data register. It is of 16 bits and is divided into two 8-bit registers DH and DL to also perform 8-bit instructions. It is used in the multiplication and input/output port addressing. Example:

MUL BX (DX, AX = AX \* BX)

**5. SP:** This is the stack pointer. It is of 16 bits. It points to the topmost item of the stack. If the stack is empty the stack pointer will be (FFFE)H. Its offset address is relative to the stack segment.

**6. BP –** This is the base pointer. It is of 16 bits. It is primarily used in accessing parameters passed by the stack. Its offset address is relative to the stack segment.

**7. SI –** This is the source index register. It is of 16 bits. It is used in the pointer addressing of data and as a source in some string-related operations. Its offset is relative to the data segment.

**8. DI –** This is the destination index register. It is of 16 bits. It is used in the pointer addressing of data and as a destination in some string-related operations. Its offset is relative to the extra segment.